N90-29033

A DESIGN PHILOSOPHY FOR MULTI-LAYER NEURAL NETWORKS WITH APPLICATIONS TO ROBOT CONTROL

Nader Vadiee and Mo Jamshidi

CAD Laboratory Systems/Robotics
Department of Electrical and Computer Engineering
The University of New Mexico
Albuquerque, New Mexico 87131
USA

ABSTRACT

A system is proposed which receives input information from many sensors that may have diverse scaling, dimension, and data representations. The proposed system tolerates sensory information with faults. The proposed self-adaptive processing technique has great promise in integrating the techniques of artificial intelligence and neural networks in an attempt to build a more intelligent computing environment. The proposed architecture can provide a detailed decision tree based on the input information, information stored in a long-term memory, and the adapted rule-based knowledge. A mathematical model for analysis will be obtained to validate the cited hypotheses . An extensive software program will be developed to simulate a typical example of pattern recognition problem. It is shown that the proposed model displays attention, expectation, spatio-temporal, and predictory behavior which are specific to the human brain. The anticipated results of this research project are (1) creation of a new dynamic neural network structure, (2) applications to and comparison with conventional multi-layer neural network stuctures such as multi-layer perceptron, neo-cognitron, etc. The anticipated benefits from this research are vast. The model can be used in a neuro-computer architecture as a building block which can perform complicated, nonlinear, time-varying mapping from a multitude of input excitory classes to an output or decision environment. It can be used for coordinating different sensory inputs and past experience of a dynamic system and actuating signals. The commercial applications of this project can be the creation of a special-purpose neuro-computer hardware which can be used in spatio-temporal pattern recognitions in such areas as air defense systems , e.g. target tracking, and recognition. Potential robotics-related applications are trajectory planning, inverse dynamics computations, hierarchical control, task-oriented control, and collision avoidance.

1. INTRODUCTION

The problem and opportunity that is discussed in this paper is the subject of current research in the field of neurocomputing. The neural networks that are in current use lack the dynamic behavior, i.e. attention, expectation, and temporal trends recognition abilities.

Adaptive Resonance Technique models such as ART 1 and ART2 , proposed by Carpenter and Grossberg [1,2] are two typical examples of neural nets with dynamic behavior. According to Rumelhart, et.al. [3] internal representations can be learned using error back propagation. Lippmann [4] gives a thorough discussion of different neural networks and their potentials. We will analyse those nets using our new approach discussed in Part 3. Self-organizing Kohonen nets [5-8] will be tested for their pattern classification capabilities for our purpose. The neural network called "neocognitron" [9-11] is used to test the ideas of clustering in the hidden layers and temporal trend analysis. One of the investigators'interest areas lies in robot systems. There has been some research undertaken in applying neural nets to the control of robot manipulators [12-14]. Bavarian [15] gives examples of the applications of neural nets to fault tolerant systems.

The investigators believe that the proposed neurocomputer and the approach unifies all these efforts and provides a general scheme for analysis and design of special-purpose neurocomputers. The type of operation that almost all living organisms or man-made engineering systems as information processors perform can be thought of as a mapping from some element in an input set χ to an element in some output set Ψ, i.e χ -- M --> Ψ.

The set χ contains elements of input information that have to be represented by some appropriate representational system, M is the process of mapping or decision making, and Ψ is the set containing possible decisions, and/or commands to be reached. The process of mapping , M, can be very complex and is based on input information, past experiences and/or a set of rules, beliefs, values, expectations criteria and constraints.

If the objects or elements in the input set can be represented by n-tuple vectors X and elements in the output set be represented by m-tuple vectors Y using some sort of sensing, coding or representational system, then M becomes a complex , time varying, nonlinear and adaptive mapping function f. Then the engineering problem in its most general form is defined by f : X(t) --> Y(t). Figure 1 shows a pictorial description of the general dynamic mapping problem. Here, sensing and preprocessing of the input information provides the system with r-classes of input excitory, i.e., X1, X2, ..., Xr. The STM (short-term memory) blocks give a dynamism to the system that enables it to derive the temporal patterns and trends by storing the recent history of the input classes Xi(t), i=1,2,...,r. This history is represented by Xi(t-ti), Xi(t-2ti), ..., Xi(t-kti) values.

Each Xi can have its own representational system and in general Xi's are vectors of ni dimension that each component is discretized by pi levels, and hence there are pi ^ ni different possible Xi points in the input subspace Si.

The block called LTM (long- term memory) may act as a rule-based data structure which can modify and facilitate the mapping function f. Y1, Y2,......, Ys represent different output classes that can be different motor signals and/or commands with some desired time sequence. Thus, the function f may depend on patterns found in input, temporal patterns in past experienced inputs, patterns in LTM, and a system of rules. The Xi's vectors belonging to different input classes may be corrupted by noise and/or damaged as a result of: ***Translation, Rotation, Distortion, Scaling, Styling, Partial occlusion and marring*** that may happen to the input sensory information. The dimension of the input vectors Xi can be excessively high.

The output Yj can be an input Xi exemplar when the above mentioned noise is filtered out, or the above mentioned damages are compensated for. The output Yj can be a set of exemplar classes, category numbers, or decisions made. The input vectors Xi can be sensory input from vision, touch, accoustic, and state-space sensors in a robotic system, and Yj's can be different actuation signals sent to actuators. Moreover, f is the required coordination between input and output. The mapping function f is the information processing operation that can be:

- Mathematical mapping approximation, developed for a funcion f : X CR^n ---> Y C R^m
- Extraction of relational knowledge from binary data bases
- Probability density function estimation
- Pattern classification
- Categorization of data
- Process of decision making
- High, medium, and low level control law executions

As far as the on-line information processing capabilities of the current analog electronic circuits and systems and the off-line information processing capabilities of modern serial computers are concerned, it seems that they have certain limits in a variety of applications. In modern conventional computers the information processing operation f has to be explicitly known, procedural, and programmable. The process is done in steps and in serial. The input information cannot undergo a great deal of distortion and damage due to noise and/or other previously mentioned causes.

A look-up table fashioned for f may not be feasible when the input vectors are of high dimension. The most important of all, the information processing operation f simply is not well known and has to be learned through experience and examples.

In summary, in cases where the input is of a very high dimension, is corrupted by noise, damaged by distortion and the function f is not explicitly known and has to be learned, adapted, or is a complex nonlinear function, then the conventional computers, analog or digital, are incapable and a new computing model and philosophy is needed.

The specific technical problem or opportunity addressed is that we believe that all the information processing carried out in most neural nets is based on this paradigm that in order to map X to Y the neural net first maps the vector X to a set of vectors ei, i = 1,2,....,r, in feature spaces F1, F2,...., Fr by feature extraction abilities of the processing nodes,. Then the feature vectors ei's are mapped to the appropriate output yi's. The idea is this : if Xi --> Yi and Xj-->Yj and we have ||yi-yj|| < d, where ||.|| is some defined norm, and d is some small positive real value, then we have : **Xi is similar to Xj**. Also we believe that if we continue the feature extraction operation by mapping X's to a set of feature spaces, then there exists a certain feature space in which two similar inputs Xi and Xj are mapped to two close or neighboring points such that $||e_{ki} - e_{kj}|| <$ **epsilon** for some 1<k<r and a small positive value epsilon.

Similarity is a relative concept that is defined with respect to a set of features found in a feature space. So we can say that similar inputs produce clusters of points in certain feature space. These clusters can be detected using self-orgainizing neural networks.

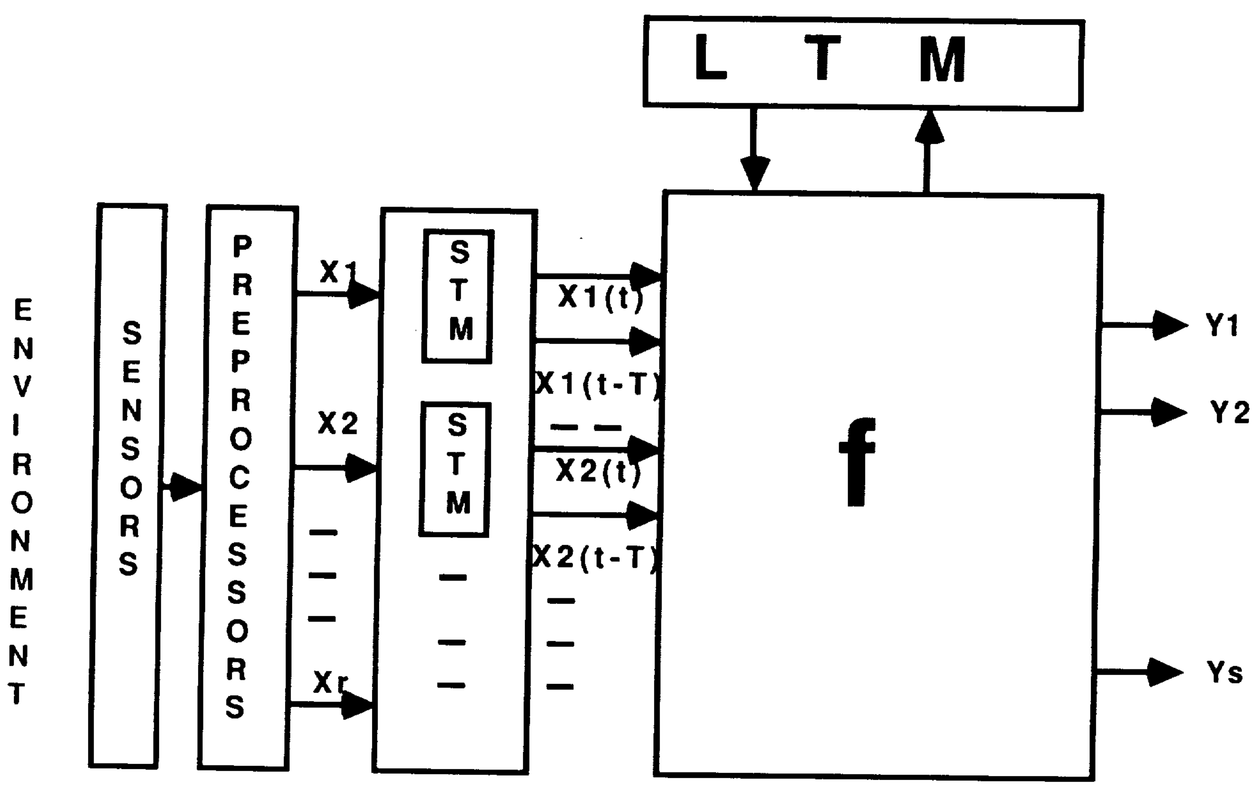

Figure 1. A Pictorial Description of the General Mapping Problem

2. PROBLEM STATEMENTS

The statement of the problems addressed here can be briefly summarized as follows :

1) Is there any clustering phenomenon occurring in the hidden layers of a multi-layer feed-forward neural network ? If so, is it possible to detect these clusterings by using some self-organizing neural nets, for example, a Kohonen map ?

2) Is it possible to recognize the temporal patterns and trends in the features using short-term memory units at the output of the above-mentioned self-organizing layers ?

3) Is it possible to continue the feature extraction and pattern classification action using the self-organizing maps' outputs augmented and made into a new input layer, and thus forming a categorizing tree describing the input ?

4) It is desirable to build neural nets in a bidirectional way to investigate the effect of priming of the former layers'nodes using the feedback from the latter layers' nodes.

5) It is interesting to prove the idea that if two inputs cluster in some hidden layer F_i, then they would cluster in all other layers F_j for any $j >= i$.

6) By simulating the visual pattern recognition net, i.e. "neocognitron", we would like to detect the dynamism in features. For example, to recognize the temporal changes in some specific features. It is possible to detect the motion of a hand-written number across the input field providing that all other features are fixed. Thus, it is possible to build temporal pattern classifiers.

7) In currently used neural networks, we start by knowing almost nothing about the mapping function f and begin the process of adaptation using a set of training points. We want to investigate a new idea that allows us to add the knowledge, if any, that we have about the mapping process to the system. This knowledge can be a set of rules from an expert. In other words, we would like to investigate a method to integrate the traditional techniques of artificial intelligence with those of artificial neural networks.

3. PROPOSED NEUROCOMPUTER ARCHITECTURE

We can simulate any unconventional transfer function from multiple inputs to multiple outputs using neural networks. The interesting point is that the transfer function can be learned by examples and experience. The inputs can be corrupted by noise and the system still remain to a high degree fault tolerant.

Traditional engineering systems extract analytical well defined features. For example, the first derivative, average, time integral, etc. In neural networks we can extract or even discover quite complicated features that are not explicitly known. Every object in the world has some specific features that distinguish it from the rest of the objects and any two objects are related from certain feature points.

Neural networks use a combination of state space transformation, feature extraction, and curvilinear transformation to reach to a canonical form for an otherwise complex mapping function. A set of objects can be classified in an infinite number of ways depending on the common defined features.

In the feed-forword multilayer perception, during the process of training by using the back propagation learning algorithm, the hidden layers are forced to form those features that would highlight the similarily between the inputs. In these nets, once the process of training is finished no further changes in the network will occur. By providing a type of feedback loop in the neural net, we are able to give the net some kind of attention and expectation abilities usually seen in natural intelligent systems. Sometimes the trend in input / output is such that certain priming of the sensory nodes or feature detection nodes can be achieved.

A network that has attention capability is able to tune on certain input patterns and ignore the others. It is able to prime the sensory nodes to make them provide more information of the kind that is most necessary for classification or mapping action. Sometimes the time sequence of output is associated with some form of time sequence of input, and certain expectancy state is aroused in the neural system. If it becomes strong enough, it may ignore and cut off the input and provide the output through the expectation mechanism. The priming action can be achieved through the inhibitory inputs of "instar" nodes or through their threshold level values.

The attention and expectation capabilities provide the net with a sort of dynamism that increases the speed of processing and saves time in computations. As far as the application to the control of robotic systems is concerned the neural nets can be used wherever a system identification or input / output approximation is required. The neural net gives the mapping between the sensory input and the controlled outputs. One can distinguish the following subsystem in a neuro-computer structure: ***1. Input layers , 2. Feature layers, 3. Kohonen layers, 4. Output layers, 5. Expectation FBK loops, 6. Attention FBK loops, 7. Temporal features detection mechanisms, 8. Short-and long-term memory blocks***.

If the nodes in the Kohonen layer, when excited, show a kind of persistence with an exponential decay function, this will provide means of studying the temporal patterns of the input sensory signals and extract the temporal features and store them in a new layer for further processing.

Computational algorithms and codes will be developed to simulate the input /output function of a single artificial neural node in all its functionalities. It is believed that by too much simplification we lose a lot of capabilities that a single processing node can have.

Typical multilayer feed-forward artificial neural nets are simulated and the idea of clustering in each layer is investigated by using Kohonen self-organizing layers, complex decision boundaries can be formed using line segments, planes or hyperplanes. The idea of using curvilinear mapping is investigated and is used in reducing the complexity of decision boundaries.

Bidirectional associative memories are implemented to study the effect of feedback paths on the performance of multilayer nets. It is believed that by using feedback paths and controlling the threshold levels in each layer we are able to give a kind of expectation and attention capabilities to neural processors. This will increase the speed of processing and save computational time . Hierarchical structures provide a kind of tree that classify the input data. The notion that Kohonen layers are the best candidate which can integrate neuro - computers with other more traditional types of computers will be investigated.

If we build the basic processing nodes in their full input/output characteristics, the decaying output will provide us means to analyze the temporal patterns. New temporal features can be extracted using Kohonen layers retain previous inputs. Relative intensily of nodes provides us a sense or means of measuring time. The training set provides stable points in each layer that make the centers of clusterings. These clusterings can be surfaced by using Kohonen layers.

If two input vectors cluster in a feature layer ej , they will cluster in feature ei for $j < i < r$. The best training set is the set providing the largest number of classes in the hidden layers. Each input vector Xi is stored in r feature vectors eij, j = 1,....,r and in r Kohonen layer vectors Kij, j = 1,...., r, and so on. By exciting any of these vectors we should be able to excite all others. It means that this structure can have mental images within the context of psychology.

The neuro - computer architecture proposed consists of successive layers of **1. Feature layers , 2. Class layers (Kohonen), (pattern classifiers), 3. Motor output layers (integrating layers).** These nets are completely bidirectional and the output of Kohonen layers can be interfaced to digital computers .

If there are n layers and p input exemplars, there will be n^p stable vectors in the layers. A new input should cluster around one of these n^p vectors. These p input exemplars should not be correlated and there should be some differences in feature. The success in convergence depends on how well these n ^ p vectors are representative of all possible future input data.

The end result of a supervised learning is producing clustering regions in the one to the last layer. The efficiency of training algorithms is studied as applicable to producing these cluster regions around the examplars. Kohonen layers, if proved useful for our purpose, can provide a means of

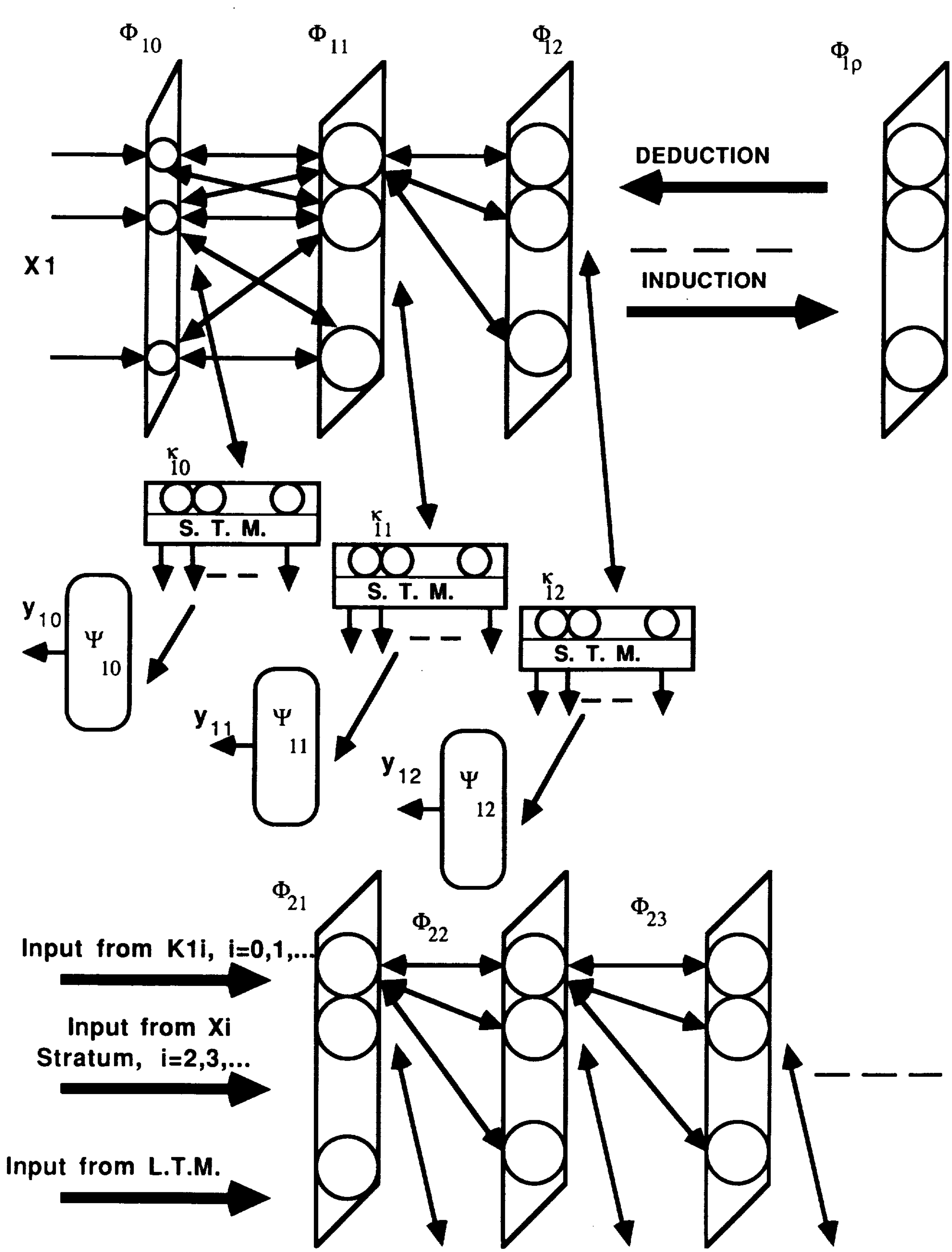

Figure 2. Proposed Module for Neuro-computer Architecture

integrating neural computers. The possibility to analyse temporal patterns provides ways for robotic system's dynamic trajectory planning.

Attention and expectation capabilities reduce the processing time in input data classifications. Figure 2 illustrates the proposed stucture for a generic subsystem for neurocomputers. In order to simplify the illustration, we concentrate on input class X1 only. Similar arguments can be applied to other input classes not shown. X1 is the input layer. O_{1j} 's are feature layers for the input class 1, k_{1j} are self-organizing layers, Y_{1j}'s are different branching outputs. The outputs from self-organizing layers can be augmented with outputs from other input classes , and outputs from LTM constitute a new higher level input layer.

Continued feature extraction in successive layers results in the formation of disjoint decision regions which are linearly discriminant. These disjoint decision regions can be detected by clustering methods using self-organizing neural networks. The clustering layers form a decision tree. This tree can be used to classify and categorize input information for future decision making.

4. FUTURE TRENDS AND POTENTIAL APPLICATIONS

The results obtained from this proposal will help understand the internal process going on in a neural network. It shows how the optimum set of distinctive features is extracted in the process of learning and how these features are used to classify the patterns. Attention, expectation, use of short-term memory, and long-term memory give a sense of dynamism that the current neural networks are lacking.

The initial phase of the research will provide the mathematical model for further design and implementation of adaptive dynamic controllers for different applications. The mathematical model and software programs developed will be used to design task-oriented high level controllers for such systems as a robot manipulator. The hardware implementation of the proposed neurocomputer will be postponed until a second phase of the project.

A neurocomputer with dynamic transfer function equipped with attention, expectation, and temporal trend analysis can be used as a special-purpose computer that can be integrated with conventional digital computers. The techniques from artificial intelligence can be used to equip the neurocomputer with rule-based intelligence. These special neurocomputers can be used in aerospace avionics, aircraft control, low-cost visual inspection systems in manufacturing, power plant fault detection, just to name a few areas of applications. This special-purpose computer can be used for many nonlinear and time-varying problems associated with a robot control system such as inverse kinematics, trajectory planning, vision, and control. These applications are being simulated in C language using a MACII computer.

5. BIBLIOGRAPHY

1. Carpenter and Grossberg, " ART 2 Self Organization of Stable Category Recognition Codes for Analog Input Patterns," Applied Optics, Dec., 1987.

2. Carpenter, G. and Grossberg, G. " The ART of Adaptive Pattern Recognition by a Self Organizing Neural Network," IEEE Computer Magazine, March 1988, pp. 77.

3. Rumelhart, D.E., Hinton, and Williams, "Learning Internal Representations by Error Propagation, " in Parallel Distributed Processing : Exploration in the Microstructure of Cognition : Foundations, D.E. Rumelhart and J. L. McClelland (eds.), Vol. 1, MIT Press, 1986.

4. Lippmann, R., "An Introduction to Computing with Neural Nets, " IEEE ASSP Magazine, pp. 4-22, April, 1987.

5. Kohenen, T., "An Introduction to Neural Computing, " Neural Networks, Vol. 1, pp. 3-16, 1988.

6. ----- , Self-organization and Associate Memory, Springer-Verlag, Berlin, 1984.

7. ----- , "Analysis of Self-organizing Process, "Biological Cybernetics, Vol. 44, pp. 135-140, 1982.

8. -----, Self-organization and Associative Memory, 2nd Edition, Springer -Verlag, Berlin, 1987.

9. Fukushima, K., and Miyake, S. , "Neocognitron: a new algorithm for pattern recognition tolerant of deformations and shifts in position," Pattern Recognition, Vol. 15, pp. 455, 1982. (computer simulations that demonstrate first order invariant in a pattern set.)

10. Fukushima, K., "A neural network model for selective attention in visual pattern recognition," Biological Cybernetics, Vol. 55, pp.5-15, 1986.

11. Fukushima, K., "A Neural Nerwork for Visual Pattern Recognition," IEEE Computer, March 1988, pp. 65.

12. Guez, A., Eilbert, J., Kam, M., "Neuromorphic Architecture for Fast Adaptive Robot Control," IEEE International Conference on Neural Networks , San Diego, CA, June, 1987.

13. Guez, A., Eilbert, J., Kam, M., "Neural Network Architecture for Control," IEEE Control Systems Magazine, April 1988, pp.22.

14. Psaltis, D., Sideris, A., and Yamamura, A., "Neural Controllers," Proc. IEEE ICNN, San Diego, CA, 1987.

15. Bavarian, B., "Introduction to Neural Networks for Intelligent Control," IEEE Control Systems Magazine, Vol. 8, No. 2, April 1988, pp. 3-7.